Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

January 9, 2020

Databases IIB: DBMS-Implementation
— Exercise Sheet 9 —

Please read Part a) and mark questions which you want to discuss in class. You only have to submit Part f) to h). Please upload your solution into the StudIP file folder called "Hausaufgabe_9" in the StudIP entry of the lecture. The deadline is January 14 (the day before the next lecture).

It is permitted to form groups of up two members, but please make sure that both members can fully explain all homeworks submitted by the group. Please upload only one file per group. Name the file such that it contains the names of all group members.

Not all submitted homeworks will be corrected, but all homework exercises will be discussed in class. If you should have questions about your homework, please ask! A precondition for getting credit for this course is that you submit solutions to two thirds of the homeworks. Obviously wrong or very incomplete submissions do not count.

Repetition Questions

a) What would you answer to the following questions in an oral exam?

- What is a "segment" in Oracle? What data can be stored in a segment?
- While the name "segment" is specific to Oracle, the task is required for any DBMS. Can a DBMS use a function of the operating system for this task? Name some advantages and disadvantages.
- Why is the relationship between tables and segments in Oracle normally "1:1", but in general "n:m"? Give at least a name of the constructs that cause exceptions to the "1:1" rule.
- If you want to program a segment manager for your own DBMS, what would be its interface?
- What is an "extent" in Oracle? Why does Oracle allocate storage in extents, and not in single blocks?
- How can a segment manager find free storage space in a file or tablespace if a segment is created or needs to grow? Oracle has two different solutions:
  - "`EXTENT MANAGEMENT DICTIONARY`" (the older version), and
  - "`EXTENT MANAGEMENT LOCAL`" (the newer version). This has variants with "`UNIFORM SIZE`" and "`AUTOALLOCATE`".

  How would you do this if you programmed a DBMS?

- How can one specify the initial amount of storage that is reserved for a table? Why can that be useful if you know or can estimate the size of the table?
- What is a ROWID or TID (tuple identifier)? What are the main components?
- A main source of ROWIDs are indexes. Give a very high level view of an index. What would be a (greatly simplified) interface of an index?
- What are the functions of the row manager in an DBMS? What are the two basic ways in which rows can be accessed?
- Discuss advantages and disadvantages of guaranteeing stable ROWIDs for the entire lifetime of a tuple.
- How can you see the ROWID of a tuple in Oracle? How can you get a readable representation of the components.
- How can you select a row with a given ROWID in Oracle?
- Explain a typical block format for a heap file, when rows can have variable length. How can one reuse the space of a deleted row for a slightly larger row if the block has in total enough space (but not in one piece)?
- What happens when a row is updated and no longer fits into its block? How is the row found with its ROWID after the update?
- What happens if the row is updated a second time and must be moved again to a new block? How many block accesses will be needed to find the row with a given ROWID?
- Why does this ROWID concept prescribe a minimum size for the rows (basically, the storage size of a ROWID)?
- Suppose that there are many migrated rows. For which operations does this influence performance? What can you do to improve performance?
- Consider an update done in a full table scan. How can the DBMS handle migrated rows, i.e. when should it update them: When the full table scan reaches the original storage place or when the full table scan reaches the current storage place? What happens if the row must be moved again?
- If you anticipate that there will be updates that increase the storage size of the rows, what should you do in Oracle?
- How can a good value for `PCTFREE` be calculated? What are the assumptions for this calculation?
- What are the consequences of a `PCTFREE` value that is too large or a value that is too small? When does it make sense to set `PCTFREE` to 0 (which is not the default value)?
- If you would develop your own DBMS, can you think of an alternative to `PCTFREE` (that needs less or other assumptions)?

Some Interesting Information

b) Have a look at at least one of the following documents:

- MySQL Internals Manual
  [https://dev.mysql.com/doc/internals/en/]
- MySQL Coding Guidelines:
  [https://dev.mysql.com/doc/dev/mysql-server/latest/]
  [PAGE_CODING_GUIDELINES.html]
  This also refers to the Google C++ Style Guide:
  [https://google.github.io/styleguide/cppguide.html]
- Oracle Database Administrator's Guide: Managing Datafiles
  [https://docs.oracle.com/cd/B28359_01/server.111/b28310/dfiles001.htm]
  This contains an explanation of absolute and relative file numbers that appear in the data dictionary tables.
- Rachid Belaid: Introduction to PostgreSQL Physical Storage.
  [http://rachbelaid.com/introduction-to-postgres-physical-storage/]
- Heikki Linnakangas: New Free Space Map and Visibility Map.
  [https://wiki.postgresql.org/images/8/81/FSM_and_Visibility_Map.pdf]

In-Class Exercises

c) Find out whether our database uses local extent management, and if yes, whether with a uniform extent size or "`AUTOALLOCATE`". You find the answer in the columns `EXTENT_MANAGEMENT` and `ALLOCATION_TYPE` in the table `DBA_TABLESPACES`. The value `SYSTEM` for `ALLOCATION_TYPE` means "`AUTOALLOCATE`". It is possible that the settings are different for different tablespaces, so you should also print the tablespace name.

d) Which different extent sizes are used in our database? Have a look at `DBA_EXTENTS`. Find a database object with large extent sizes and list all its extents ordered by `EXTENT_ID`.

e) How much free space is there in each tablespace or each data file? There are different approaches to get this information, check whether they return the same values. For instance, you can add the segment size for each segment in a tablespace and substract it from the tablespace size. You can also add the extent sizes in each data file and subtract it from the data file size. You can also have a look at `DBA_FREE_SPACE`.

Homework Exercises

f) Select any three questions from a) and write answers to them. I hope that not all students select the first three items on the list. Please write in total at least 150 words (if you write less than ten lines, I might have to count the words).

g) Consider the following relation:

```
R(A: numeric(5), B: varchar(10), C: varchar(50))
```

Suppose that `PCTFREE` is 10, and one inserts 10 000 tuples of the form

```
(99999, null, 'abcdefghijklmnopqrstuvwxyz')
```

Answer the following questions:

- What is the storage size of the number `99999` in Oracle? (Do not count the column length byte.)
- How long is a row of this example table in Oracle?
- How many bytes are reserved in each block by the `PCTFREE` setting? (Assume a block size of 8 KByte, i.e. 8192 Byte).
- How many tuples fit into one block? (Use 90 Bytes as the overhead for the block header.)
- How many blocks will be used for the table? (Do not forget the segment header block.)

Do all these calculations theoretically. We will do a practical test in the next lab session.

h) Suppose an update sets the attribute `B` for all tuples to `'ABCDEFGHIJ'`. Will there be migrated rows? Answer this question theoretically:

- How long will the rows be after the update, i.e. how many additional bytes will be needed?
- Is the `PCTFREE` space reserve sufficient for the growth of all rows stored in one block?
- If not, what would have been a good value for `PCTFREE` to avoid migrated rows for this update?